Testing Booleanity and the Uncertainty Principle

Tom Gur and Omer Tamuz*

March 2, 2013

Abstract

Let $f : \{-1,1\}^n \to \mathbb{R}$ be a real function on the hypercube, given by its discrete Fourier expansion, or, equivalently, represented as a multilinear polynomial. We say that it is Boolean if its image is in $\{-1,1\}$.

We show that every function on the hypercube with a sparse Fourier expansion must either be Boolean or far from Boolean. In particular, we show that a multilinear polynomial with at most k terms must either be Boolean, or output values different than -1 or 1 for a fraction of at least $2/(k+2)^2$ of its domain.

It follows that given black box access to f, together with the guarantee that its representation as a multilinear polynomial has at most k terms, one can test Booleanity using $O(k^2)$ queries. We show an $\Omega(k)$ queries lower bound for this problem.

We also consider the problem of deciding if a function is Boolean, given its explicit representation as a k term multilinear polynomial. The naïve approach of evaluating it at every input has $O(kn2^n)$ time complexity. For large k (i.e, exponential) we present a simple randomized $O(kn\sqrt{2^n})$ algorithm. For small k we show how the problem can be solved deterministically in $O(k^3n)$.

Our proofs crucially use Hirschman's entropic version of Heisenberg's uncertainty principle.

*Weizmann Institute of Science. E-mail: {tom.gur,omer.tamuz}@weizmann.ac.il. Supported by ISF grant 1300/08. Omer Tamuz is a recipient of the Google Europe Fellowship in Social Computing, and this research is supported in part by this Google Fellowship.

1 Introduction

Let f be a function from $\{-1,1\}^n$ to $\mathbb{R}$. Equivalently, one can consider functions on $\{0,1\}^n$ or $\mathbb{Z}_2^n$, as we do below. A natural way to represent such a function is as a multilinear polynomial. For example:

$$f(x_1,x_2,x_3) = x_1 - 2x_2x_3 + 3.5x_1x_2.$$

This representation is called the *Fourier expansion* of f and is extremely useful in many applications (cf., [13]). The coefficients of the Fourier expansion of f are called the *Fourier transform* of f. We denote the Fourier transform by $\hat{f}$, and think of it too as a function from $\{-1,1\}^n$ to $\mathbb{R}$.

We say that f is Boolean if $f(x)=1$ or $f(x)=-1$ for all x in its domain. An interesting question in the field of discrete Fourier analysis of Boolean functions is the following: what does the fact that f is Boolean tell us about its Fourier transform $\hat{f}$? Is there a simple characterization of functions that are the Fourier transform of Boolean functions?

We propose the following observation that lies at the basis of our proofs: f is Boolean if and only if the convolution (over $\mathbb{Z}_2^n$) of $\hat{f}$ with itself is equal to the delta function. This follows from the convolution theorem, as we show below in Claim 3.1.

Equipped with this characterization, we consider two variations of the question of determining whether or not f is Boolean. First, we consider the case that we are given black box access to a function f, together with the guarantee that its representation as a multilinear polynomial has at most k terms. We show that $O(k^2)$ queries to f suffice to answer this question correctly with high probability. This follows from the following combinatorial result: in Theorem 1.1 we show that if f is not Boolean then it is not Boolean for at least a $2/(k+2)^2$ fraction of its domain. More generally, we show that for any set $D \subset \mathbb{R}$ of size d, either the image of f is contained in D, or else $f(x) \notin D$ for at least a $d!/(k+d)^d$ fraction of the domain of f. We prove an $\Omega(k)$ lower bound for this problem.

Booleanity testing bears resemblance to problems of property testing of functions on the hypercube (see, e.g., [3, 5, 6, 12]). See Section 1.4 below for further discussion.

We also consider the question of determining if f is Boolean, given its explicit representation as a multilinear polynomial with k terms. The naïve algorithm that evaluates f at all possible inputs takes $O(kn2^n)$, assuming that arithmetic operations over $\mathbb{R}$ take constant time.

We show that, for small k, the problem can be solved deterministically in time complexity $O(k^3n)$. For large k we show that the problem can be answered correctly with high probability in $O(kn\sqrt{2^n})$. In the extreme case that $k=\Theta(2^n)$ - i.e., almost all the Fourier coefficients are non-zero - then the naïve algorithm has time complexity $O(k^2\log k)$, whereas our algorithms runs in $O(k^{3/2}\log k)$.

Our proofs rely on the discrete version of *Heisenberg's uncertainty principle*, which can be viewed as an analogue of the *Schwartz-Zippel lemma.* We are not familiar with other applications of it in Computer Science, but expect that more can be found, in particular in cryptography. See Sections 1.3 and 1.5 below for further discussion.

In the following Section 1.1 we present our main results, and in Sections 1.2, 1.3, 1.4 and 1.5 we elaborate on the background and relation to other work. Section 2 contains formal definitions, and proofs appear in Section 3.

1.1 Main results

Let $\mathcal{F}_k$ be the family of functions $f : \{-1,1\}^n \to \mathbb{R}$ that can be represented as a multilinear polynomial with at most k terms. Recall that we say that f is Boolean if its image is contained in $\{-1,1\}$.

The following theorem is a combinatorial result, stating that a function with a sparse Fourier expansion is either Boolean or far from Boolean.

Theorem 1.1. *Every function $f \in \mathcal{F}_k$ is either Boolean, or satisfies*

$$\mathbb{P}_x\left[f(x) \notin \{-1,1\}\right] \geq \frac{2}{(k+2)^2}$$

where $\mathbb{P}_x[\cdot]$ denotes the uniform distribution over the domain of f.

We in fact prove a more general result:

Theorem 1.2. *Let $D \subset \mathbb{R}$ be a set with d elements. Then for any $f \in \mathcal{F}_k$ one of the following holds.*

- *Either $\mathbb{P}_x[f(x) \in D] = 1$,*
- *or $\mathbb{P}_x[f(x) \notin D] \geq \frac{d!}{(k+d)^d}$,*

where $\mathbb{P}_x[\cdot]$ denotes the uniform distribution over the domain of f.

That is, either f's image is in D, or it is far from being in D. In particular, for $D = \{-1,1\}$ (or $\{0,1\}$, or any other set of size two), this theorem reduces to Theorem 1.1

An immediate consequence of Theorem 1.1 is the following result.

Theorem 1.3. *For every $\epsilon > 0$ there exists a randomized algorithm with time complexity $O(k^2 \log(1/\epsilon))$ that, given k and oracle access to a function $f \in \mathcal{F}_k$,*

- *returns* true *if f is Boolean, and*
- *returns* false *with probability at least $1-\epsilon$ if f is not Boolean.*

This result can easily be extended to test whether the image of a function on the hypercube is contained in any finite set, using Theorem 1.2.

We prove the following lower bound:

Theorem 1.4. *Let A be an algorithm that, given k and oracle access to a function $f \in \mathcal{F}_k$,*

- *returns* true *if f is Boolean, and*
- *returns* false *with probability at least $2/3$ if f is not Boolean.*

Then A has time complexity $\Omega(k)$.

We next turn to the problem of deciding if f is Boolean given its Fourier expansion, or its representation as a multilinear polynomial.

Theorem 1.5. *For every $\epsilon > 0$ there exists a randomized algorithm with time complexity $O\left(kn\sqrt{2^n}\log(1/\epsilon)\right)$ that, given the multilinear polynomial representation of a function $f \in \mathcal{F}_k$,*

- *returns* true *if f is Boolean, and*
- *returns* false *with probability at least $1-\epsilon$ if f is not Boolean.*

We here think of k as large - at least of the order of $2^{n/3}$. Assuming $k = 2^n$ (the largest it can be) the size of the input to the algorithm is $O(k)$. The naïve algorithm of evaluating f at every possible input has time complexity $O\left(k^2 \log k\right)$. We reduce this to $O\left(k^{3/2}\log k\right)$.

For small k (i.e., $k < 2^{n/3}$), we show a *deterministic* algorithm that outperforms the previous algorithm.

Theorem 1.6. *There exists a deterministic algorithm with time complexity $O(k^3 n)$ that, given the multilinear polynomial representation of a function $f \in \mathcal{F}_k$, returns* true *if f is Boolean and* false *otherwise.*

As before, we note that these results can be extended to test other images, using the same techniques.

1.2 The Fourier transform of Boolean functions

Let f, g be functions from $\mathbb{Z}_2^n$ to $\mathbb{R}$. Their convolution $f * g$ is a also a function from $\mathbb{Z}_2^n$ to $\mathbb{R}$ defined by

$$[f * g](x) = \sum_{y \in \mathbb{Z}_2^n} f(y) g(x + y),$$

where the addition "$x + y$" is done using the group operation of $\mathbb{Z}_2^n$. The commonly used definition of convolution evaluates g at $x - y$, whereas this one does so at $x + y$. But since $x + x = 0$ for all $x \in \mathbb{Z}_2^n$ then $x + y = x - y$, and thus this definition coincides with the usual one.

An observation that lies at the basis of our proofs is a characterization of the Fourier transforms of Boolean functions: $\hat{f} : \mathbb{Z}_2^n \to \mathbb{R}$ is the Fourier transform of a Boolean function if its convolution with itself is equal to the delta function:

$$\hat{f} * \hat{f} = \delta.$$

This is our Claim 3.1; it follows from the convolution theorem. Equivalently, given a function f on $\mathbb{Z}_2^n$, one can shift it by acting on it with $x \in \mathbb{Z}_2^n$ by $[xf](y) = f(x + y)$. Hence the observation above can be stated as follows: If and only if a function is orthogonal to its shifted self, for all non-zero shifts in $\mathbb{Z}_2^n$, then it is the Fourier transform of a Boolean function.

1.3 The uncertainly principle

A distribution over a discrete domain S is often represented as a non-negative function $f : S \to \mathbb{R}^+$ which is normalized in L_1, i.e., $\sum_{x \in S} f(x) = 1$.

In Quantum Mechanics the state of a particle on a domain S is represented by a *complex* function on S, and the probability to find the particle in a particular $x \in S$ is equal to $|f(x)|^2$. Accordingly, f is normalized in L_2, so that $\sum_{x \in S} |f(x)|^2 = 1$.

Often, the domain S is taken to be $\mathbb{R}$ (or some power thereof). In this continuous case one represents the state of a particle by a function $f : \mathbb{R} \to \mathbb{C}$ such that $\int_{x \in \mathbb{R}} |f(x)|^2 dx = 1$, and then $|f(x)|^2$ is the probability density function of the distribution of the particle's position. The Fourier transform of f, denoted by $\hat{f}$, is then also normalized in L_2 (if one chooses the Fourier transform operator to be unitary), and $|\hat{f}(x)|^2$ is the probability density function of the *distribution of the particle's momentum*.

The Heisenberg uncertainty principle states that the variance of a particle's position times the variance of its momentum is at least one - under an appropriate choice of units. Besides its physical significance, this is also a purely mathematical statement relating a function on $\mathbb{R}$ to its Fourier transform.

Hirschman [10] conjectured in 1957 a stronger entropic form, namely

$$H_e\left[f\right] + H_e\left[\hat{f}\right] \geq 1 - \ln 2,$$

where $H_e\left[f\right] = -\int_{x \in \mathbb{R}} |f(x)|^2 \ln |f(x)|^2 dx$ is the differential entropy of f. This was proved nearly twenty years later by Beckner [1].

When the domain S is $\mathbb{Z}_2^n$ (equivalently, $\{-1,1\}^n$) then a similar inequality holds, but with a different constant. Let $f : \mathbb{Z}_2^n \to \mathbb{C}$ have Fourier transform $\hat{f} : \mathbb{Z}_2^n \to \mathbb{C}$, and normalize $\|f\|_2 = \|\hat{f}\|_2 = 1$. Then

$$H\left[f\right] + H\left[\hat{f}\right] \geq n,$$

where $H\left[f\right] = -\sum_{x\in\mathbb{Z}_2^n} |f(x)|^2 \log_2 |f(x)|^2$.

1.4 Relation to property testing

We note that the problem of testing Booleanity is similar in structure to a property testing problem. Since its introduction in the seminal paper by Rubinfeld and Sudan [15], property testing has been studied extensively, both due to its theoretical importance, and the wide range of applications it spanned (cf. [7, 8]). In particular, property testing of functions on the hypercube is an active area of research [3, 5, 6, 12].

A typical formulation of property testing is as follows: Given a fixed property P and an input f, a property tester is an algorithm that distinguishes with high probability between the case that f satisfies P, and the case that f is ϵ-far from satisfying it, according to some notion of distance.

The algorithm we present for testing Booleanity given oracle access is similar to a property testing algorithm. However, in our case there is no proximity parameter: we show that if a function is not Boolean then it *must* be far from Boolean, and can therefore be proved to not be Boolean by a small number of queries. This type of property testing algorithms have appeared in the context of the study of adaptive versus non-adaptive testers [9].

1.5 Discussion and open questions

In this paper we use a discrete entropy uncertainty principle to both prove a combinatorial statement concerning functions on the hypercube, and to construct algorithms for computational problems. To the best of our knowledge, this is the first time this tool has been used in the context of theoretical computer science.

The discrete uncertainty principle is, in a sense, a dual to the *Schwartz-Zippel* lemma [17, 16]: both limit the number of roots of a polynomial, given that it is sparse. Given the usefulness of the Schwartz-Zippel lemma, we suspect that more combinatorial applications can be found for the discrete uncertainty principle.

For example, Biham, Carmeli and Shamir [2] show that an RSA dechiperer who uses hardware that has been maliciously altered can be vulnerable to an attack resulting in the revelation of the private key. The assumption is that the dechiperer is not able to discover that it is using faulty hardware, because the altered function returns a faulty output for only a very small number of inputs. The uncertainty principle shows that such malicious alteration is impossible to accomplish with succinctly represented functions: when the Fourier transform of a function is sparse then it is impossible to "hide" elements in its image.

As for the scope of this study, many questions still remains open. In particular, there is a gap between the lower bound and the upper bound for testing Booleanity with oracle access; we are disinclined to guess which of the two is not tight.

2 Definitions

The following definitions are mostly standard. We deviate from common practice by considering both a function and its Fourier transform to be defined on the same domain, namely $\mathbb{Z}_2^n$. Some readers might find $\{0,1\}^n$ or $\{-1,1\}^n$ a more familiar domain for a function, and likewise the power set of $[n]$ a more familiar domain for its Fourier transform.

Denote $\mathbb{Z}_2 = \mathbb{Z}/2\mathbb{Z}$. For $x, y \in \mathbb{Z}_2^n$ we denote by $x + y$ the sum using the $\mathbb{Z}_2^n$ group operation. The equivalent operation in $\{-1,1\}^n$ is pointwise multiplication.

Let $f : \mathbb{Z}_2^n \to \mathbb{R}$. We denote its L_2-norm by

$$\|f\| = \sqrt{\sum_{x \in \mathbb{Z}_2^n} f(x)^2}, \tag{1}$$

denote its support by

$$\operatorname{supp} f = \{x \in \mathbb{Z}_2^n \ : \ f(x) \neq 0\}, \tag{2}$$

and denote its entropy by

$$H[f] = -\sum_{x \in \mathbb{Z}_2^n} f(x)^2 \log f(x)^2, \tag{3}$$

where logarithms are base two and $0 \log 0 = 0$, by the usual convention is this case.

We call a function $f : \mathbb{Z}_2^n \to \mathbb{R}$ *Boolean* if its image is in $\{-1,1\}$, i.e., if $f(x) \in \{-1,1\}$ for all $x \in \mathbb{Z}_2^n$.

Let $\hat{f} : \mathbb{Z}_2^n \to \mathbb{R}$ denote the *discrete Fourier transform* (also known as the *Walsh-Fourier transform* and *Hadamard transform*) of f, or its representation as a multilinear polynomial:

$$\hat{f}(x) = \frac{1}{2^n} \sum_{y \in \mathbb{Z}_2^n} f(y)\chi_y(x), \tag{4}$$

where the characters χ_y are defined by

$$\chi_y(x) = \begin{cases} 1 & \sum_{i:y_i=1} x_i = 1 \\ -1 & \text{otherwise} \end{cases}.$$

Note that the sum $\sum_{i:y_i=1} x_i$ is over $\mathbb{Z}_2$ and that x_i, y_i are (respectively) the i'th coordinate of x and y. It follows that the *discrete Fourier expansion* of f is

$$f(x) = \sum_{y \in \mathbb{Z}_2^n} \hat{f}(y)\chi_y(x). \tag{5}$$

We denote by $\mathcal{F}_k$ the family of functions $f : \mathbb{Z}_2^n \to \mathbb{R}$ for which $|\operatorname{supp} \hat{f}| \leq k$.

We define $\delta : \mathbb{Z}_2^n \to \mathbb{R}$ by

$$\delta(x) = \begin{cases} 1 & \text{when } x = (0, \ldots, 0) \\ 0 & \text{otherwise} \end{cases}.$$

If we denote by $\mathbf{1}(x) : \mathbb{Z}_2^n \to \mathbb{R}$ the constant function such that $\mathbf{1}(x) = 1$ for all $x \in \mathbb{Z}_2^n$, then it is easy to verify that

$$\hat{\mathbf{1}} = \delta. \tag{6}$$

Given functions $f, g : \mathbb{Z}_2^n \to \mathbb{R}$, their *convolution* $f * g$ is also a function from $\mathbb{Z}_2^n$ to $\mathbb{R}$, defined by

$$[f * g](x) = \sum_{y \in \mathbb{Z}_2^n} f(y)g(x+y). \tag{7}$$

We denote

$$f^{(2)} = f * f,$$

and more generally $f^{(k)}$ is the convolution of f with itself k times. $f^{(0)}$ is taken to equal δ, since $f * \delta = f$.

3 Results

3.1 The Fourier transform of Boolean functions

The convolution theorem (see., e.g., [11]) for $\mathbb{Z}_2^n$ states that the Fourier transform of the pointwise multiplication of two functions is equal to the convolution of their Fourier transforms. Since the Fourier transform operator is its own inverse - up to a constant - then it follows that the Fourier transform of the convolution is equal to the pointwise product of the Fourier transforms - up to a constant:

$$\widehat{f \cdot g} = \hat{f} * \hat{g}, \qquad \text{and} \qquad \widehat{f * g} = 2^n \hat{f} \cdot \hat{g}. \tag{8}$$

The correctness of the constants can be verified by, for example, setting $f = g = \mathbf{1}$.

Claim 3.1. *$f : \mathbb{Z}_2^n \to \mathbb{R}$ is Boolean iff $\hat{f} * \hat{f} = \delta$.*

Proof. Recall that a function is Boolean if its image is in $\{-1, 1\}$. Hence $f : \mathbb{Z}_2^n \to \mathbb{R}$ is Boolean iff $f^2 = \underline{1}$. Applying the Fourier transform to both sides yields $\widehat{f^2} = \delta$, by Eq. 6. By Eq. 8 we have that $\widehat{f^2} = \hat{f} * \hat{f}$, and the claim follows. □

More generally, let f be such that its image is in $D = \{y_1, \dots, y_d\}$. Then $(f - y_1) \cdots (f - y_d) = 0$, and so

$$\left(\hat{f} - y_1\delta\right) * \cdots * \left(\hat{f} - y_d\delta\right) = 0. \tag{9}$$

3.2 The discrete uncertainty principle

The discrete uncertainty principle for $\mathbb{Z}_2^n$ is the following.

Theorem 3.2. *For any $f : \mathbb{Z}_2^n \to \mathbb{R}$ such that $\|f\| > 0$ it holds that*

$$H\left[\frac{f}{||f||}\right] + H\left[\frac{\hat{f}}{||\hat{f}||}\right] \geq n. \tag{10}$$

Proof. Let U be a unitary n by n matrix such that $\max_{ij} |u_{ij}| = M$. Let $x \in \mathbb{C}^n$ be such that $\|x\| > 0$. Then Theorem 23 in Dembo, Cover and Thomas [4] states that

$$H\left[\frac{x}{||x||}\right] + H\left[\frac{Ux}{||Ux||}\right] \geq 2\log(1/M).$$

Let F be the matrix representing the Fourier transform operator on $\mathbb{Z}_2^n$. Note that by our definition in Eq. 4, the transform operator F is not unitary. However, if we multiply it by $\sqrt{2^n}$ (i.e., normalize the characters χ_y) then it becomes unitary. The normalized matrix elements (which are equal to the elements of the normalized characters χ_y), are all equal to $\pm 1/\sqrt{2^n}$. Hence $M = 1/\sqrt{2^n}$, and

$$H\left[\frac{f}{||f||}\right] + H\left[\frac{Ff}{||Ff||}\right] \geq 2\log(1/M) = n.$$

□

Note that equality is achieved for any character χ_y, or alternatively for a Fourier transform of a character, which is a shifted delta function; in this case the entropy of the character is n and the entropy of its transform is zero. It has, in fact, been shown that equality is achieved only in these cases [14].

A distribution supported on a set of size k has entropy at most $\log k$, as can be shown by calculating its Kullback-Leibler divergence from the uniform distribution. Hence any distribution with entropy $\log k$ has support of size at least k. This fact, together with the discrete uncertainty principle, yields the following claim.

Claim 3.3. *For any $f : \mathbb{Z}_2^n \to \mathbb{R}$ such that $\|f\| > 0$ it holds that*

$$|\operatorname{supp} f| \cdot |\operatorname{supp} \hat{f}| \geq 2^n. \tag{11}$$

Proof. By Theorem 3.2 we have that

$$H\left[\frac{f}{||f||}\right] + H\left[\frac{\hat{f}}{||\hat{f}||}\right] \geq n.$$

Since $\log|\operatorname{supp}(f)| = \log|\operatorname{supp}(f/||f||)| \geq H[f/||f||]$ then

$$|\operatorname{supp} f| \cdot |\operatorname{supp} \hat{f}| \geq 2^n,$$

□

It follows that

$$\max\{|\operatorname{supp} f|, |\operatorname{supp} \hat{f}|\} \geq \sqrt{2^n}. \tag{12}$$

3.3 Testing Booleanity of a function given as a polynomial

Before proposing our algorithm for testing Booleanity we state and prove the following standard claim, which relates the support of functions f and g with the support of their convolution.

Claim 3.4. *Let $g, f : \mathbb{Z}_2^n \to \mathbb{R}$. Then*

$$\operatorname{supp} f * g \subseteq \operatorname{supp} f + \operatorname{supp} g.$$

Here $\operatorname{supp} f + \operatorname{supp} g$ is the set of elements of $\mathbb{Z}_2^n$ that can be written as the sum of an element in $\operatorname{supp} f$ and an element in $\operatorname{supp} g$.

Proof. Let $x \in \operatorname{supp} f * g$. Then, from the definition of convolution, there exist y and z such that $f(y) \neq 0$, $g(z) \neq 0$ and $x = y + z$. Hence $x \in \operatorname{supp} f + \operatorname{supp} g$. □

Remark 3.5. *Note that $|\operatorname{supp} f + \operatorname{supp} g|$ is at most $|\operatorname{supp} f| \cdot |\operatorname{supp} g|$. Since the straightforward calculation of $[f*g](x)$ using Eq. 7 takes $O(|\operatorname{supp} f| \cdot n)$ then calculating all of $f*g$ takes $O(|\operatorname{supp} f|^2 \cdot |\operatorname{supp} g| \cdot n)$. In particular, the calculation of $f * f$ takes $O(|\operatorname{supp} f|^3 \cdot n)$.*

We are now ready to propose the following algorithm for testing if a function $f : \mathbb{Z}_2^n \to \mathbb{R}$ is Boolean, given its representation as a multilinear polynomial, i.e., given $\hat{f}$ with support size k. We here think of k as large - at least $2^{n/3}$. Arithmetic operations over the reals are assumed to take unit time.

Theorem 3.6. *Given $f \in \mathcal{F}_k$, Algorithm 1 has time complexity $O\left(kn\sqrt{2^n}\log(1/\epsilon)\right)$. Furthermore, if f is Boolean then the algorithm returns "true", and if f is not Boolean then it returns "false" with probability at least $1 - \epsilon$.*

Algorithm 1

```
for $i \in \{1, \dots, \sqrt{2^n}\log(1/\epsilon)\}$ do
  Draw $x$ uniformly at random from $\mathbb{Z}_2^n$
  Calculate $f(x)$ and $\hat{f}^{(2)}(x)$
  if $f(x) \notin \{-1,1\}$ or $\hat{f}^{(2)}(x) - \delta(x) \neq 0$ then
    return "false"
  end if
end for
return "true"
```

Theorem 1.5 is a direct consequence of this theorem.

Proof. Given $\hat{f}$ with support of size k, calculating $f(x)$ takes $O(kn)$, as does the calculation of $\hat{f}^{(2)}(x)$, using Eqs. 5 and 7, respectively, since the sums need only be taken over the support of $\hat{f}$ (see Remark 3.5). Hence the total time complexity is indeed $O\left(kn\sqrt{2^n}\log(1/\epsilon)\right)$.

If f is Boolean, then $f(x) \in \{-1,1\}$ for all x, and by Claim 3.1 it holds that $\hat{f}^{(2)}(x) = \delta(x)$ for all x. Hence the algorithm will return "true".

If f is not Boolean, then $f(x)^2 - 1 \neq 0$ for some x, and so $\|f^2 - 1\| > 0$. Since

$$\widehat{f^2 - 1} = \hat{f}^{(2)} - \delta$$

then by Eq. 12 we have that

$$\max\left\{|\operatorname{supp} f^2 - 1|, |\operatorname{supp} \hat{f}^{(2)} - \delta|\right\} \geq \sqrt{2^n}.$$

Therefore, at each iteration, the algorithm will return "false" with probability at least $1/\sqrt{2^n}$, and the theorem follows since we repeat this for $\sqrt{2^n}\log(1/\epsilon)$ iterations. $\square$

This algorithm can be straightforwardly extended to testing whether the image of f is in some set $D = \{y_1, \dots, y_d\}$ of size d. The number of iterations is identical, but at each iteration the test is different: The first test is replaced with checking if $f(x)$ is in D, and the second is replaced with checking if

$$\left(\hat{f}(x) - y_1\delta(x)\right) * \cdots * \left(\hat{f}(x) - y_d\delta(x)\right) = 0,$$

as suggested by Eq. 9.

The support of $\hat{f} - y_i\delta$ is at most $k+1$, and so the support of the first $d/2$ terms of the expression above is of size $O(k^{d/2})$, by Claim 3.4. The same holds for the last $d/2$ terms.

Since convolution is associative then we can calculate the convolution of the first $d/2$ terms, calculate the convolution of the last $d/2$ terms, and then convolve the results, making altogether d convolutions, each of which takes $O(k^{d/2}n)$. Hence this calculation takes $O(k^{d/2}nd)$, and the algorithm's time complexity is $O\left(k^{d/2}dn\sqrt{2^n}\log(1/\epsilon)\right)$.

For $k < 2^{n/3}$ there exists a faster, deterministic algorithm to decide if f is Boolean: given $\hat{f}$, calculate $\hat{f} * \hat{f}$ and compare it to δ. Correctness follows directly from Claim 3.1. This can be done in $O(k^3 n)$ (see Remark 3.5), and so Theorem 1.6 follows. For $k < 2^{n/3}$, this algorithm is faster than Algorithm 1.

3.4 Testing Booleanity given oracle access

In this section we consider a function $f \in \mathcal{F}_k$ to which we are given oracle access. We are asked to determine if it is Boolean, or more generally if its image is in some small set D. We here think of k as being small - say polynomial in n.

We first prove the following combinatorial result:

Theorem (1.2). *Let $D \subset \mathbb{R}$ be a set with d elements. Then for any $f \in \mathcal{F}_k$ one of the following holds.*

- *Either $\mathbb{P}_x\left[f(x) \in D\right] = 1$,*
- *or $\mathbb{P}_x\left[f(x) \notin D\right] \geq \frac{d!}{(k+d)^d}$,*

where $\mathbb{P}_x\left[\cdot\right]$ denotes the uniform distribution over the domain of f.

Proof. Let $D = \{y_1, \dots, y_d\}$. Denote

$$g = \prod_{i=1}^{d}(f - y_i),$$

so that $g(x) = 0$ iff $f(x) \in D$. Then

$$\hat{g} = \left(\hat{f} - y_1\delta\right) * \cdots * \left(\hat{f} - y_d\delta\right) = \hat{f}^{(d)} + a_{d-1}\hat{f}^{(d-1)} + \cdots a_1\hat{f} + a_0\delta,$$

for some coefficients $a_0, \dots, a_{d-1}$. Therefore

$$\operatorname{supp}\hat{g} \subseteq \bigcup_{i=0}^{d}\operatorname{supp}\hat{f}^{(i)}. \tag{13}$$

We show that $|\operatorname{supp}\hat{g}| \leq (k+d)^d/d!$. Let $A = \operatorname{supp}\hat{f} \cup \{0\}$. Then by Claim 3.4 $\operatorname{supp}\hat{f}^{(i)}$ is a subset of $iA = A + \cdots + A$, where the sum is taken i times; this is the set of elements in $\mathbb{Z}_2^n$ that can be written as a sum of i elements of A. Hence

$$\operatorname{supp}\hat{g} \subseteq A \cup 2A \cup \cdots \cup dA.$$

Since $0 \in A$, then for all $i \leq d$ we have that $iA \subseteq dA$. Hence

$$\operatorname{supp}\hat{g} \subseteq dA.$$

Therefore $\operatorname{supp}\hat{g}$ is a subset of the set of elements that can be written as the sum of at most d elements of A. This number is bounded by the number of ways to choose d elements of A with replacement, disregarding order. Hence

$$|\operatorname{supp}\hat{g}| \leq \binom{|A| - 1 + d}{d} \leq \frac{(k+d)^d}{d!}, \tag{14}$$

since $|A| \leq |\operatorname{supp}\hat{f}| + 1 = k + 1$.

Now, if $f(x) \in D$ then clearly $\mathbb{P}_x\left[f(x) \in D\right] = 1$. Otherwise, $g(x)$ is different than zero for some x, and so $\|g\| > 0$. Hence we can apply Claim 3.3 and

$$|\operatorname{supp} g| \cdot |\operatorname{supp}\hat{g}| \geq 2^n.$$

By Eq. 14 this implies that

$$|\operatorname{supp} g| \geq \frac{2^n d!}{(k+d)^d}.$$

Since the support of g is precisely the set of $x \in \mathbb{Z}_2^n$ for which $f(x) \neq D$ then it follows that

$$\mathbb{P}_x\left[f(x) \notin D\right] \geq \frac{d!}{(k+d)^d}.$$

$\square$

A consequence is that a function that is not Boolean (i.e., the case $D = \{-1, 1\}$) is not Boolean over a fraction of at least $2/(k+2)^2$ of its domain. Theorem 1.3 is a direct consequence of this result: the algorithm samples f at random $\frac{1}{2}(k+2)^2 \log(1/\epsilon)$ times, and therefore will discover an x such that $f(x) \notin \{-1, 1\}$ with probability at least $1 - \epsilon$ - unless f is Boolean.

While we were not able to show a tight lower bound, we show that any algorithm would require at least $\Omega(k)$ queries to perform this task.

Proof of Theorem 1.4. Let A be an algorithm that is given oracle access to a function $f : \mathbb{Z}_2^n \to \mathbb{R}$, together with the guarantee that $\operatorname{supp} \hat{f} \leq k$. When f is Boolean then A returns "true". When f is not Boolean then f returns "false" with probability at least $1 - \epsilon$. We show that A makes $\Omega(k)$ queries to f.

Denote by B_k the set of Boolean functions that depend only on the first $\log k$ coordinates. Denote by C_k the set of functions that likewise depend only on the first $\log k$ coordinates, return values in $\{-1, 1\}$ for some $k-1$ of the k possible values of the first $\log k$ coordinates, but otherwise return 2. Note that functions in both B_k and C_k have Fourier transforms of support of size at most k.

We use Yao's Minmax principle in order to prove a lower bound for a randomized algorithm. For this purpose, we present two distributions: one for which the algorithm should return "false" (denoted by $\mathcal{D}_0$) and another for which the algorithm should return "true" (denoted by $\mathcal{D}_1$). We prove that any randomized algorithm which performs at most $o(k)$ queries would not be able to distinguish between the two distributions with non-negligible probability. This proves the claim.

Let $\mathcal{D}_1$ be the uniform distribution over B_k, and let $\mathcal{D}_0$ be the uniform distribution over C_k. Observe that an arbitrary query to f in either distribution would output a non-Boolean value with probability at most $1/k$, independently of previous queries with different values of the first $\log k$ coordinates. Therefore any algorithm that performs $o(k)$ queries would find an input for which $f(x) = 2$ with probability $o(1)$, and would therefore be unable to distinguish between $\mathcal{D}_0$ and $\mathcal{D}_1$ with noticeable probability. $\square$

4 Acknowledgments

The authors would like to thank Elchanan Mossel for a helpful initial discussion of the problem and for suggesting the application to property testing. We would like to thank Adi Shamir for suggesting the relevance to cryptography. In addition we would like to thank Ran Raz, Uriel Feige, Moni Naor and Oded Goldreich for useful discussions, and the organizers of the Marne-la-Vallée winter school on discrete Fourier analysis for providing the setting in which this study was conceived.

References

[1] W. Beckner. Inequalities in fourier analysis. *The Annals of Mathematics*, 102(1):159–182, 1975.

[2] E. Biham, Y. Carmeli, and A. Shamir. Bug attacks. *Advances in Cryptology–CRYPTO 2008*, pages 221–240, 2008.

[3] E. Blais. Testing juntas nearly optimally. In *STOC*, pages 151–158, 2009.

[4] A. Dembo, T. Cover, and J. Thomas. Information theoretic inequalities. *Information Theory, IEEE Transactions on*, 37(6):1501–1518, 1991.

[5] E. Fischer, G. Kindler, D. Ron, S. Safra, and A. Samorodnitsky. Testing juntas. In *FOCS*, pages 103–112, 2002.

[6] E. Fischer, E. Lehman, I. Newman, S. Raskhodnikova, R. Rubinfeld, and A. Samorodnitsky. Monotonicity testing over general poset domains. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, STOC '02, pages 474–483, New York, NY, USA, 2002. ACM.

[7] O. Goldreich. A brief introduction to property testing. In *Property testing*. Springer-Verlag, Berlin, Heidelberg, 2010.

[8] O. Goldreich, S. Goldwasser, and D. Ron. Property testing and its connection to learning and approximation. In *FOCS*, pages 339–348, 1996.

[9] M. Gonen and D. Ron. On the benefits of adaptivity in property testing of dense graphs. *Algorithmica*, 58(4):811–830, 2010.

[10] I. Hirschman. A note on entropy. *American journal of mathematics*, 79(1):152–156, 1957.

[11] Y. Katznelson. *An introduction to harmonic analysis*. Cambridge Univ Pr, 2004.

[12] I. Newman and T. A. Queries. Testing of function that have small width branching programs. In *Proc. of 41 th FOCS*, pages 251–258, 2000.

[13] R. O'Donnell. Analysis of boolean functions. `http://analysisofbooleanfunctions.org/`, 2012.

[14] M. Özaydin and T. Przebinda. An entropy-based uncertainty principle for a locally compact abelian group. *Journal of Functional Analysis*, 215(1):241–252, 2004.

[15] R. Rubinfeld and M. Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM J. Comput.*, 25(2):252–271, 1996.

[16] J. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *Journal of the ACM (JACM)*, 27(4):701–717, 1980.

[17] R. Zippel. An explicit separation of relativised random polynomial time and relativised deterministic polynomial time. *Information processing letters*, 33(4):207–212, 1989.